

**AMENDMENTS TO THE CLAIMS**

*Please amend the Claims as follows:*

Claims 1-29 (previously cancelled)

**30. (Currently Amended)** A computer-based method for validating a fragment of ~~a structured~~ an XML document, said computer-based method implemented in computer readable program code, said computer readable program code stored in computer memory, said computer-based method comprising steps of:

(a) receiving as input a fragment of ~~a structured~~ an XML document into a runtime validation engine;

(b) outputting a validation pass message based on an LR parsing technique as follows:

(i) obtaining a first token from said fragment of said ~~structured~~ XML document,

(ii) determining whether said first token is of element type said fragment of said ~~structured~~ XML document that is to be validated against, and if so,

(iii) obtaining next token from said fragment of said ~~structured~~ XML document,

(iv) checking whether said next token signifies end of said fragment of said structured document, and if so,

identifying a FOLLOW token from a type-mapping table for said element type in (b)(ii);

based on said FOLLOW token, performing a low level

validation using a generic XML parser and performing a high level validation using an XML schema validation parser; and

\_\_\_returning a validation pass if both validations of said generic XML parser and XML schema validation parser are successful and an annotated automaton encoding (AAE) stack is empty; and

if said next token does not signify end of said fragment of said ~~structured~~-XML document, continuing validation as in validating an entire ~~structured~~-XML document, and when successfully validated as in an entire ~~structured~~-XML document, returning to step (iii) until end of said ~~structured~~-XML document token is received and outputting a validation pass when AAE stack is empty.

**31. (Previously Presented)** The computer-based method of claim 30, wherein, when first token is not of said element type, or when said continued validation as in validating an entire document fails in step (iv) or when said AAE stack is not empty, said method returns a validation failure message.

**32. (Canceled)** The computer-based method of claim 30, wherein said structured document is an XML document.

**33. (Previously Presented)** The computer-based method of claim 30, wherein said first or next token is either an element type name or an attribute name.

**34. (Original)** The computer-based method of claim 30, wherein said first or next token is a lexeme, said lexeme being any of the following: a start tag name, an attribute name, or an end tag name.

**35. (Original)** The computer-based method of claim 30, wherein said computer-based method is implemented in conjunction with a database.

**36. (Currently Amended)** A computer-based method for validating a fragment of ~~a structured~~ an XML document, said computer-based method implemented in computer readable program code, said computer readable program code stored in computer memory, said computer-based method comprising steps of:

(a) receiving as input a fragment of an XML document into a runtime validation engine;

(b) outputting a validation pass message based on an LR parsing technique as follows:

- (i) obtaining a first token from said fragment of said XML document,
- (ii) determining whether said first token is of element type said fragment of said XML document that is to be validated against, and if so,

- (iii) obtaining next token from said fragment of said XML document,
- (iv) checking whether said next token signifies end of said fragment of said XML document, and if so,
  - identifying a FOLLOW token from an annotation record for said element type in (b)(ii);
  - based on said FOLLOW token, performing a low level validation using a generic XML parser and performing a high level validation using an XML schema validation parser; and
  - returning a validation pass if both validations of said generic XML parser and XML schema validation parser are successful and an annotated automaton encoding (AAE) stack is empty; and

if said next token does not signify end of said fragment of said XML document, continuing validation as in validating an entire XML document, and when successfully validated as in an entire XML document, returning to step (iii) until end of said fragment of said XML document token is received,

wherein, when first token is not of said element type, or when said continued validation as in validating an entire document fails in step(iv) or when said AAE stack is not empty, said method returns a validation failure message.

**37. (Original)** The computer-based method of claim 36, wherein said first or next token is either an element type name or an attribute name.

**38. (Original)** The computer-based method of claim 36, wherein said first or next token is a lexeme, said lexeme being any of the following: a start tag name, an attribute name, or an end tag name.

**39. (Original)** The computer-based method of claim 36, wherein said computer-based method is implemented in conjunction with a database.

**40. (Currently Amended)** An article of manufacture comprising a computer usable medium having computer readable program code embodied therein which implements a computer-based method for validating a fragment of ~~a structured~~ an XML document, said computer-based method implemented in computer readable program code, said computer readable program code stored in computer memory, said computer usable medium comprising:

(a) computer readable program code aiding in receiving as input a fragment of a ~~structured~~ an XML document into a runtime validation engine;

(b) computer readable program code aiding in outputting a validation pass message based on an LR parsing technique as follows:

(i) computer readable program code aiding in obtaining a first token from said fragment of said ~~structured~~ XML document,

- (ii) computer readable program code determining whether said first token is of element type said fragment of said ~~structured~~-XML document that is to be validated against, and if so,
- (iii) computer readable program code aiding in obtaining next token from said fragment of said ~~structured~~-XML document,
- (iv) computer readable program code checking whether said next token signifies end of said fragment of said ~~structured~~ XML document, and if so,
  - identifying a FOLLOW token from an annotation record for said element type in (b)(ii);
  - based on said FOLLOW token, performing a low level validation using a generic XML parser and performing a high level validation using an XML schema validation parser; and
  - returning a validation pass if both validations of said generic XML parser and XML schema validation parser are successful and an annotated automaton encoding (AAE) stack is empty; and

if said next token does not signify end of said fragment of said ~~structured~~-XML document, computer readable program code continuing validation as in

validating an entire ~~structured~~-XML document, and when successfully validated as in an entire ~~structured~~-XML document, computer readable program code returning to step (iii) until end of said ~~structured~~-XML document token is received and outputting a validation pass when AAE stack is empty.

**41. (Previously Presented)** The article of manufacture of claim 40, wherein, when first token is not of said element type, or when said continued validation as in validating an entire document fails in step (iv) or when said AAE stack is not empty, computer readable program code returns a validation failure message.

**42. (Canceled)** The article of manufacture of claim 40, wherein said structured document is an XML document.

**43. (Previously Presented)** The article of manufacture of claim 40, wherein said first or next token is either an element type name or an attribute name.

**44. (Original)** The article of manufacture of claim 40, wherein said first or next token is a lexeme, said lexeme being any of the following: a start tag name, an attribute name, or an end tag name.